Git & github

Hadley Wickham
@hadleywickham
Chief Scientist, RStudio

February 2015

Overview

- Focus mostly on the why, not the how
- Learning git & github is definitely frustrating at first
- Two big pay offs: **safety** & **community**
- Mechanics at http://r-pkgs.had.co.nz/git.html

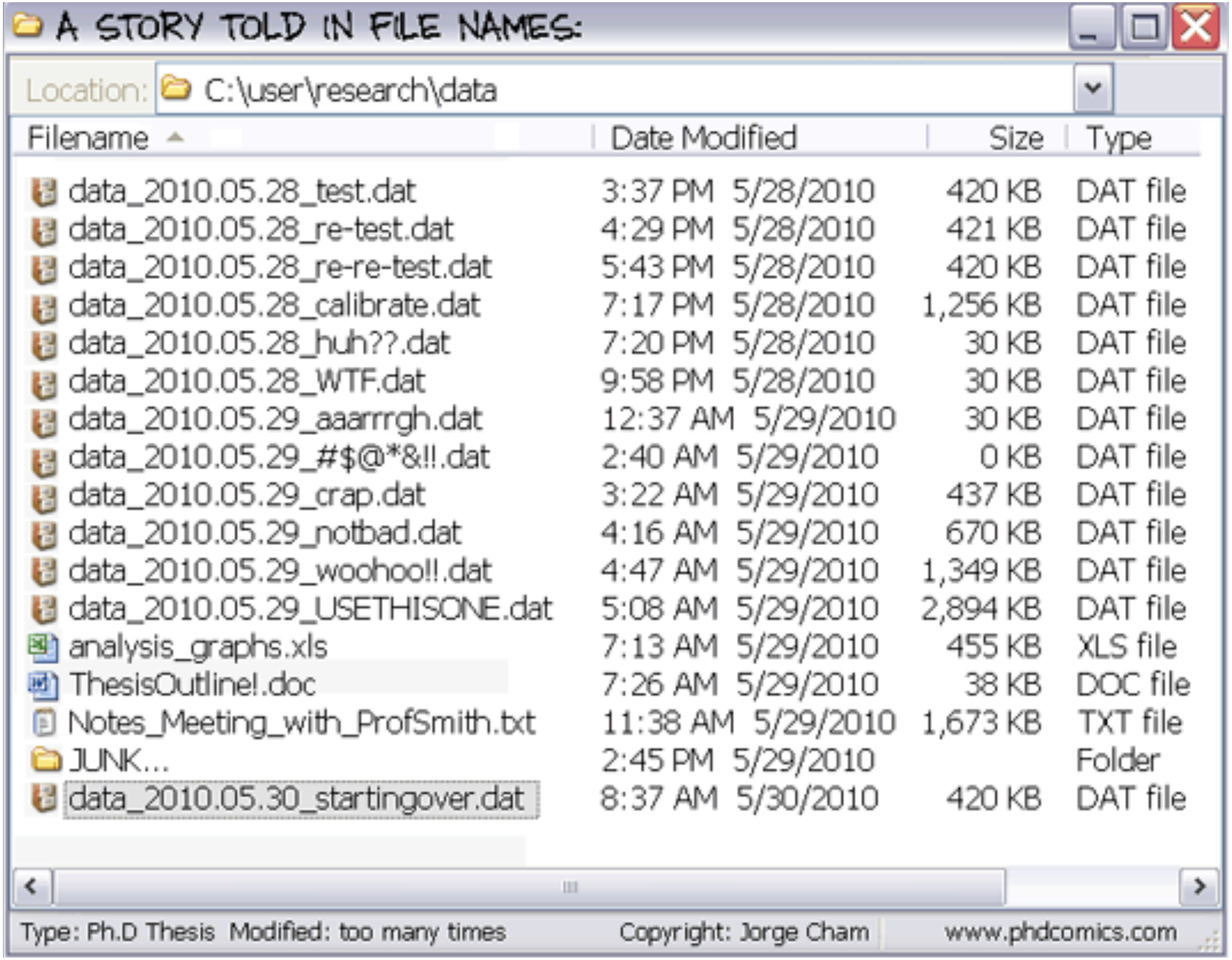
Safety

“Failures, repeated failures, are finger posts on the road to achievement. One fails forward toward success.”

— C. S. Lewis

https://www.flickr.com/photos/magnezja/8587678264

https://www.flickr.com/photos/subflux/509039029



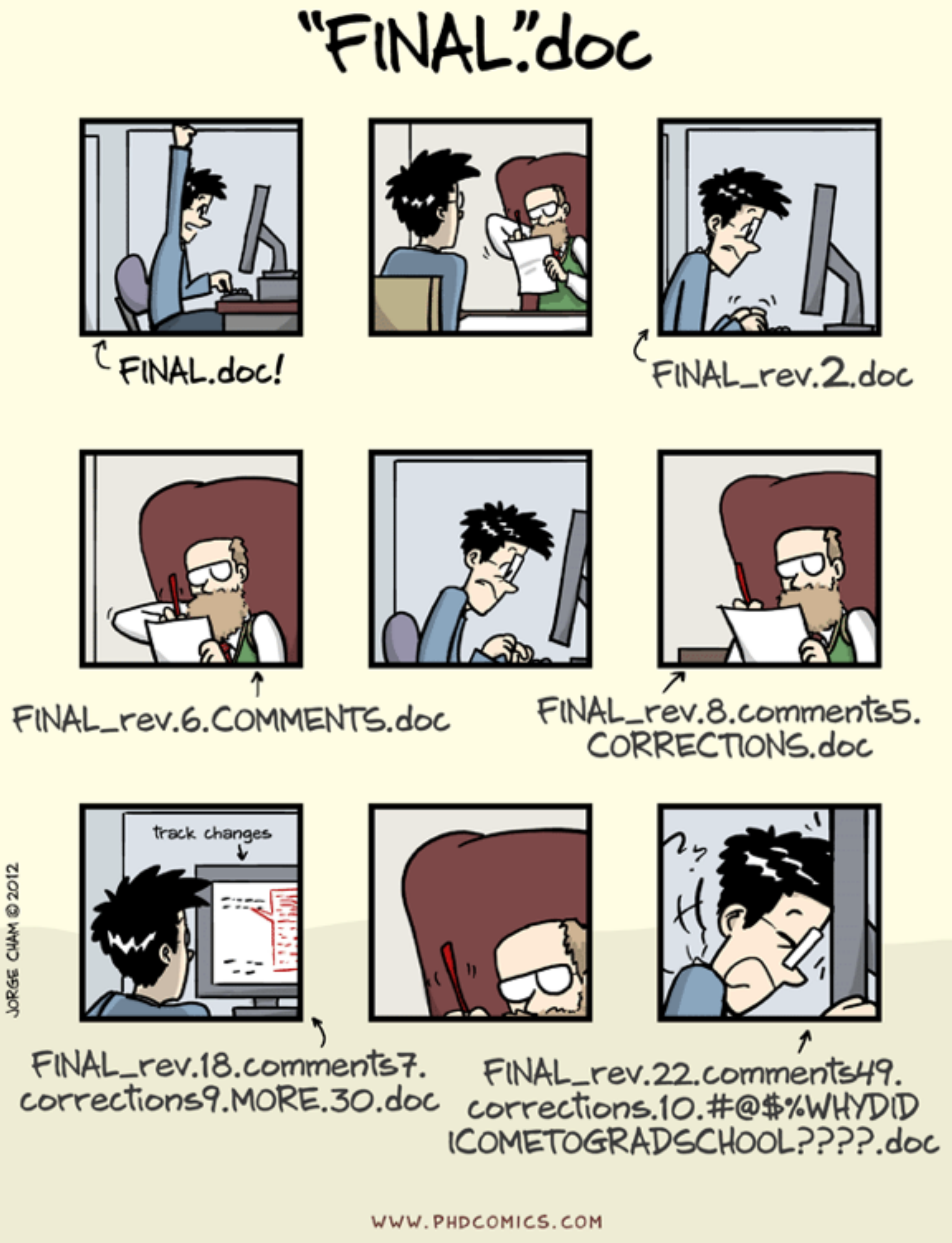
A STORY TOLD IN FILE NAMES:

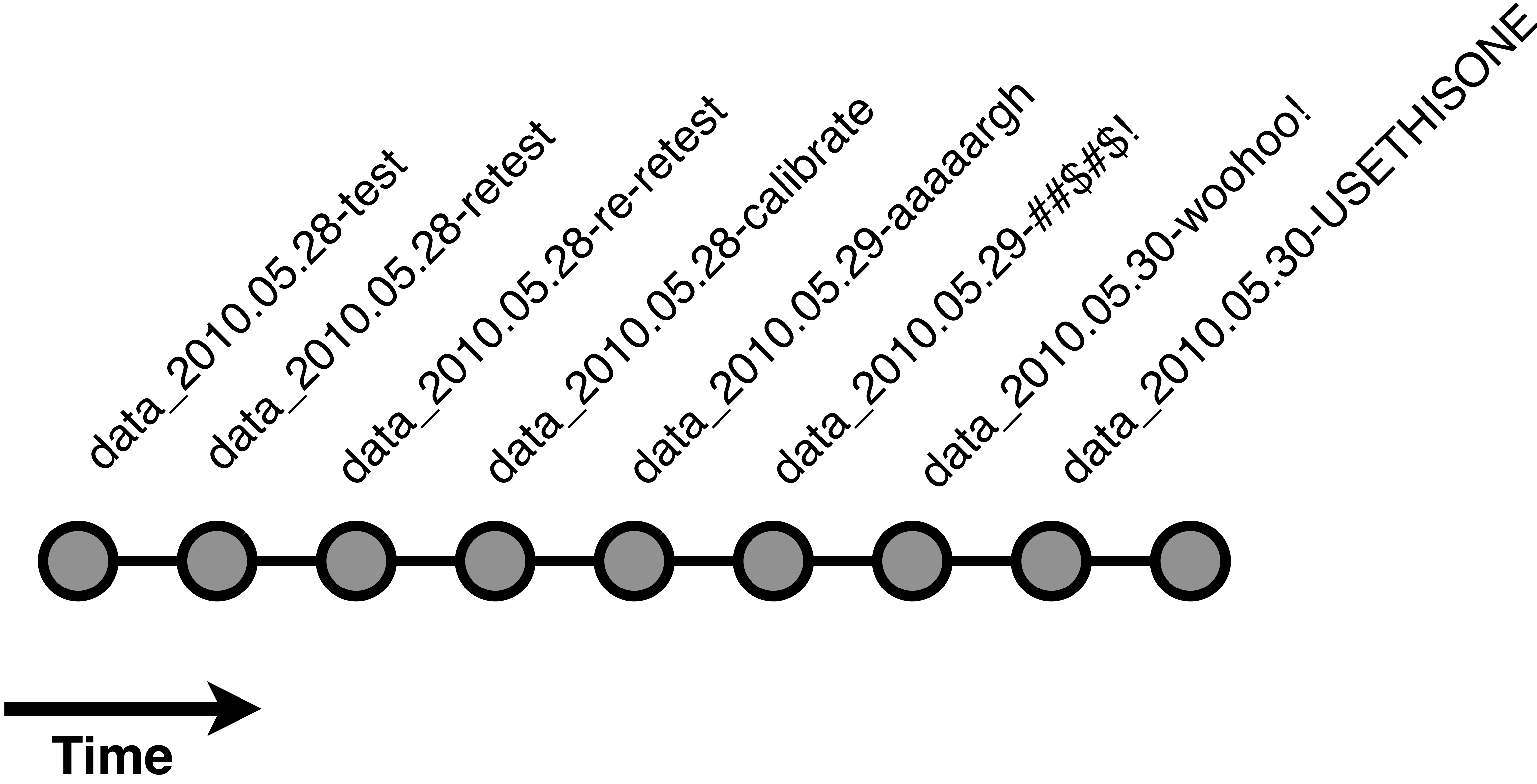
Location: C:\user\research\data

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#$@*&!!.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

http://www.phdcomics.com/comics/archive.php?comicid=1323


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM


data_2010.05.28-test
data_2010.05.28-retest
data_2010.05.28-re-retest
data_2010.05.28-calibrate
data_2010.05.29-aaaaaargh
data_2010.05.29-##$#$!
data_2010.05.30-woohoo!
data_2010.05.30-USETHISONE
Time

The history of a project can be viewed as a series of changes

Changes

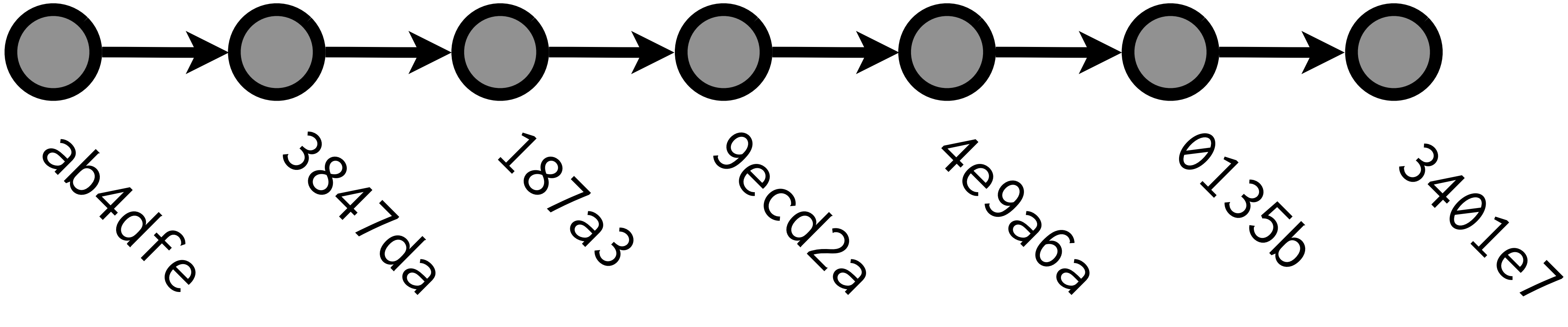
- A unique identifier
- What changed?
- When did it change?
- Who changed it?
- Why did it change?

Changes

- **A unique identifier**
- What changed?
- **When did it change?**
- Who changed it?
- **Why did it change?**

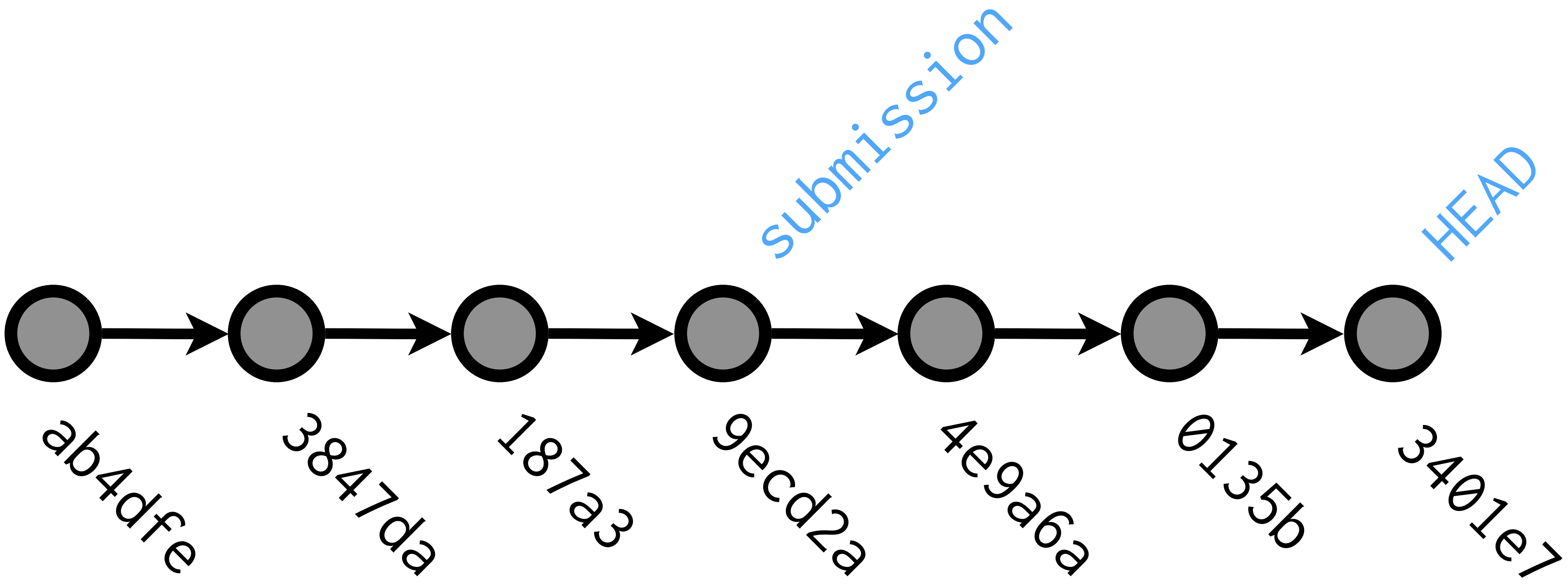
(And this obviously gets hard to coordinate for multiple files)

With git, each change (**commit**) is given a unique identifier, called a **sha**



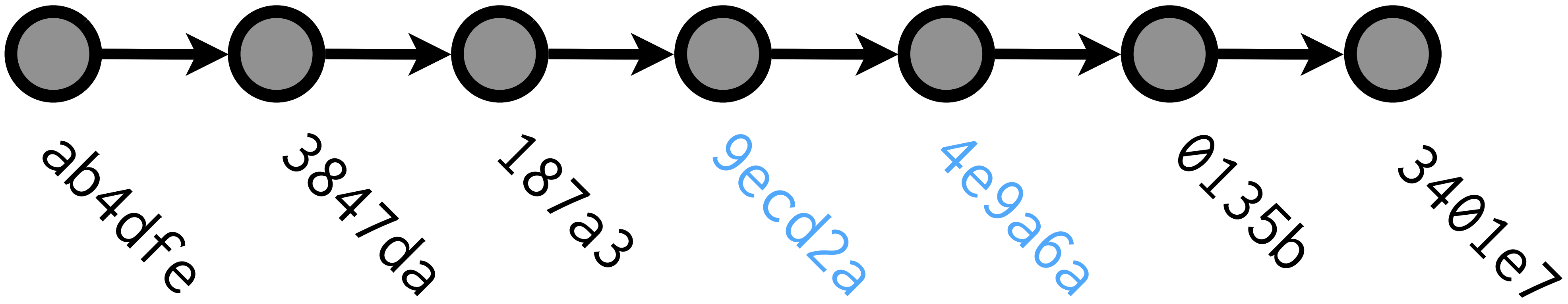
The sha is a key into a database that provides the author, date, and a description

You can also name individual commits



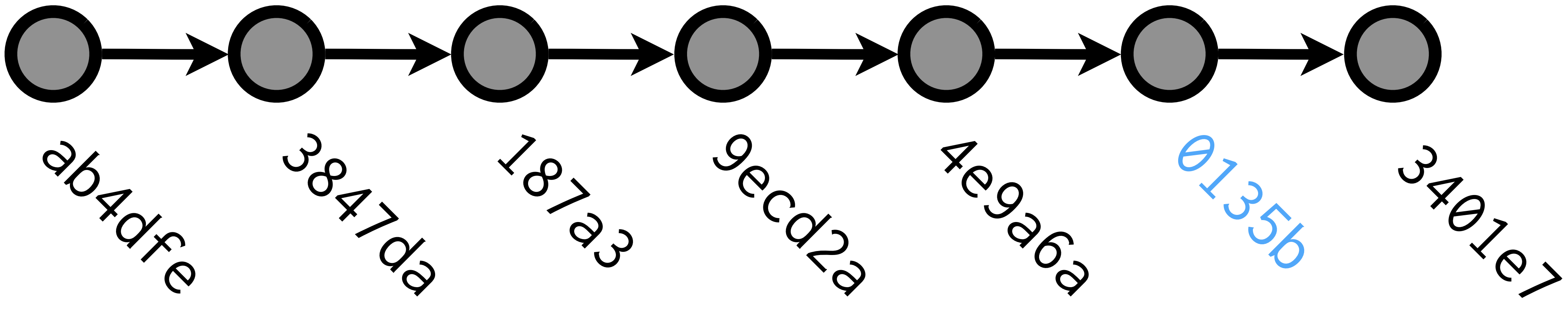
```
git tag submission 9ecd2a
```

Then see exactly what’s changed



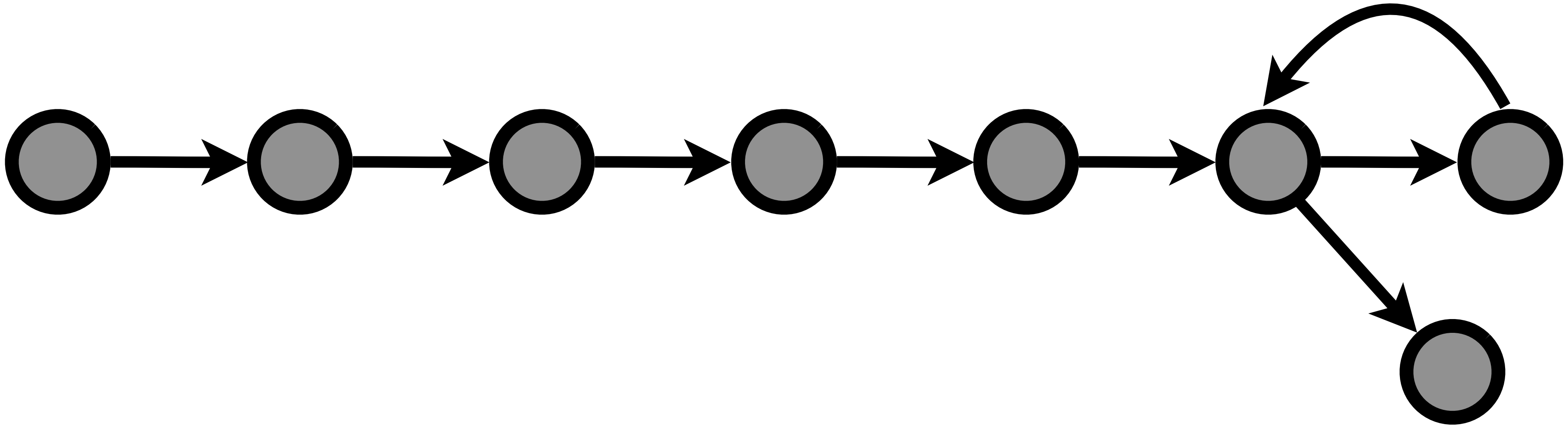
```
git diff 9ecd2a..4e9a6a
```

You can revert to a previous change with git checkout

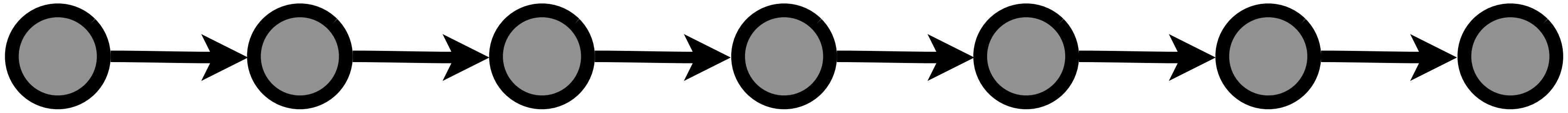


```
git checkout 0135b
```

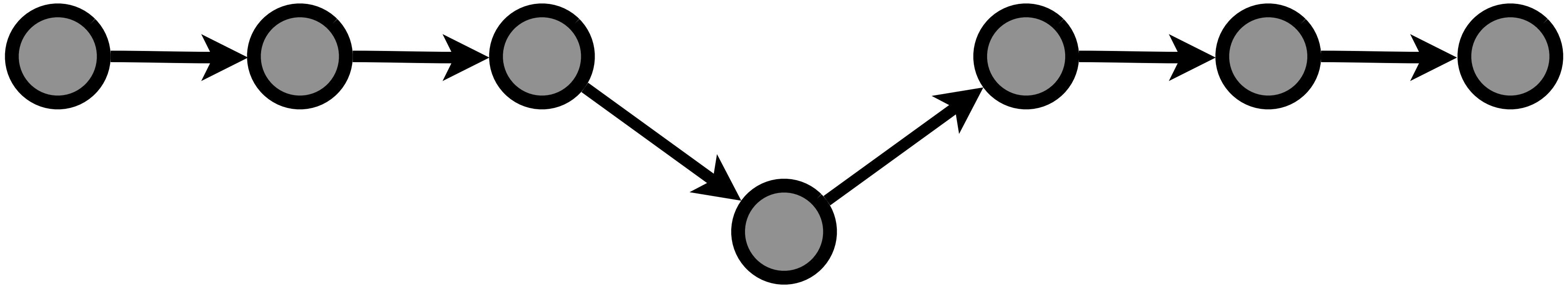
That allows you to undo mistakes



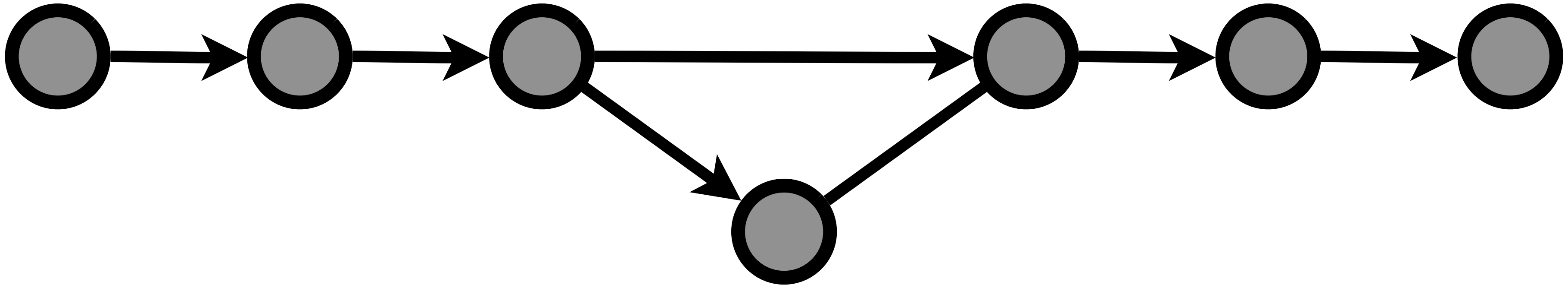
Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed. **That's risky!**



```
git rebase ...
```

Demo

http://r-pkgs.had.co.nz

http://github.com/hadley/r-pkgs

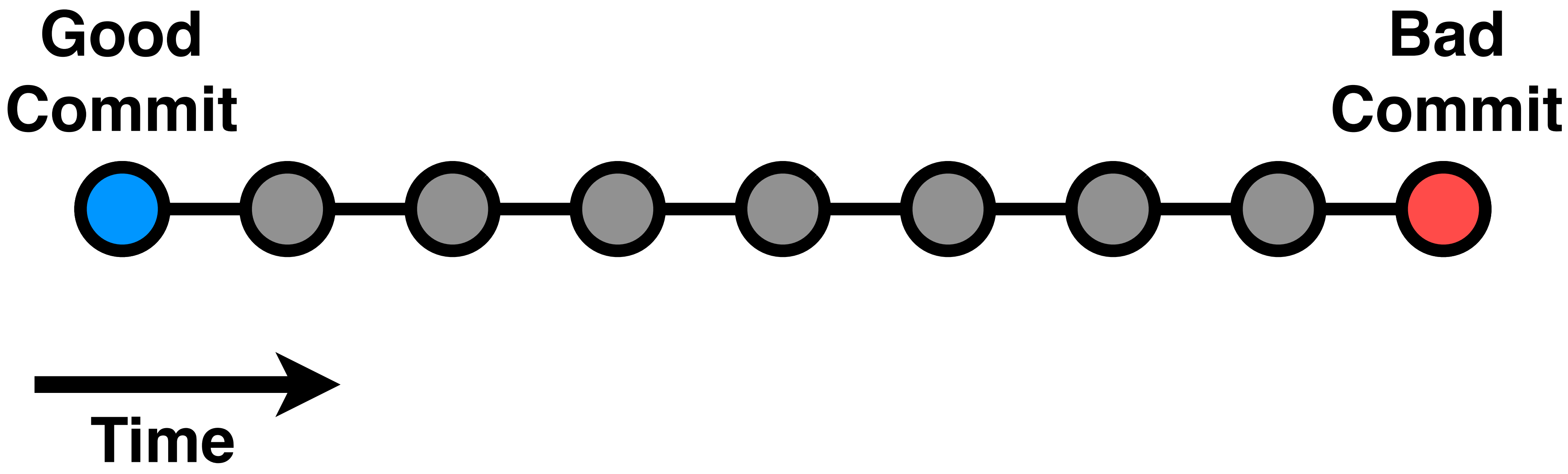
Demo

Vows package & roxygen2

Other benefits

git bisect

https://github.com/wch/bisectr

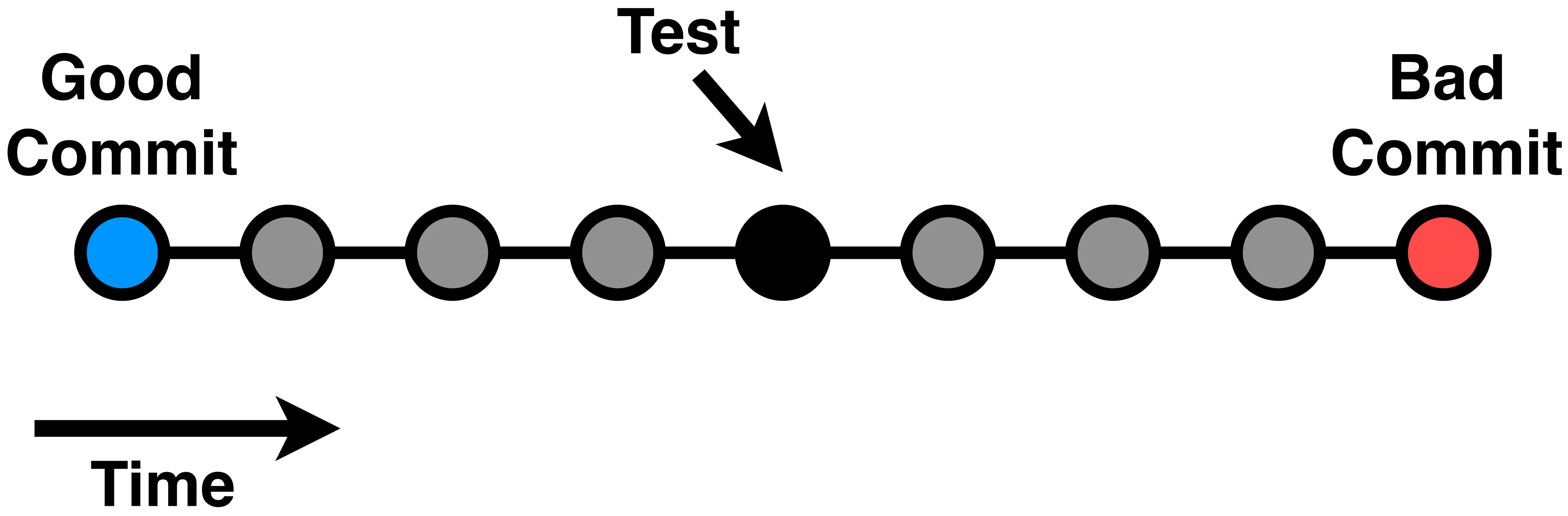


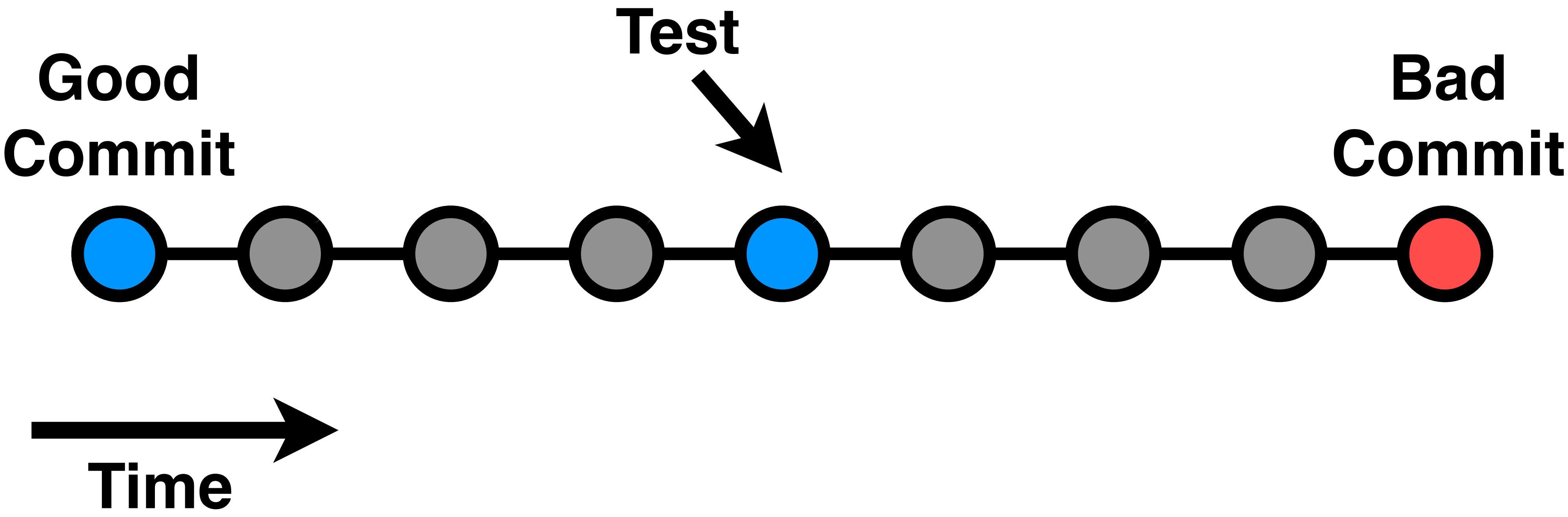
Good Commit

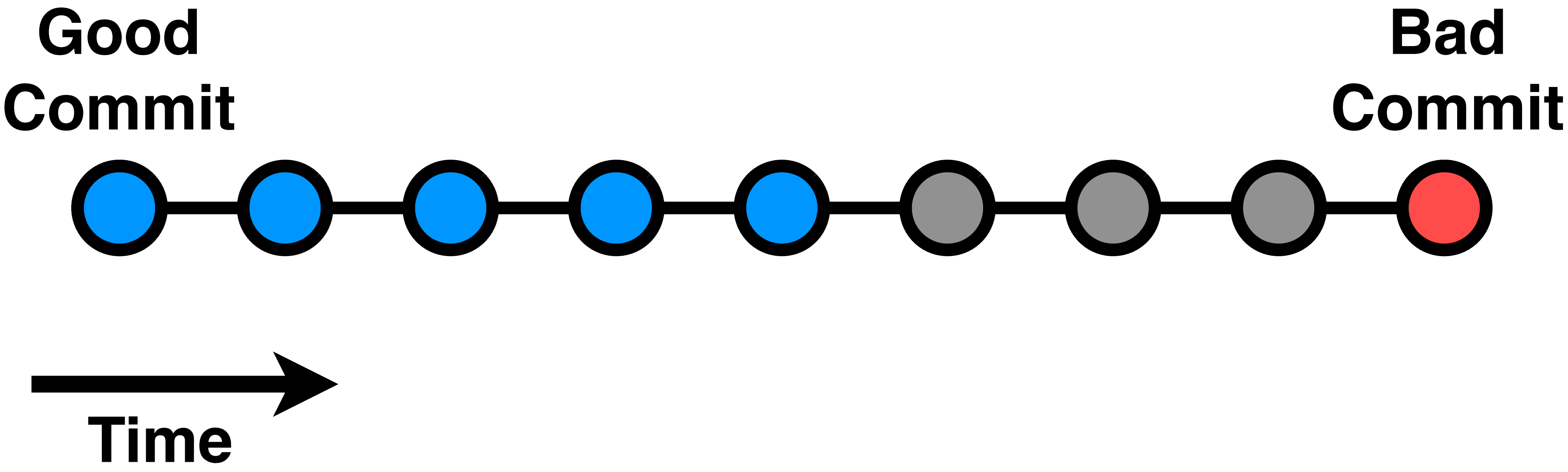
Bad Commit

Time

Where did the problem happen?


Test
Good Commit
Bad Commit
Time

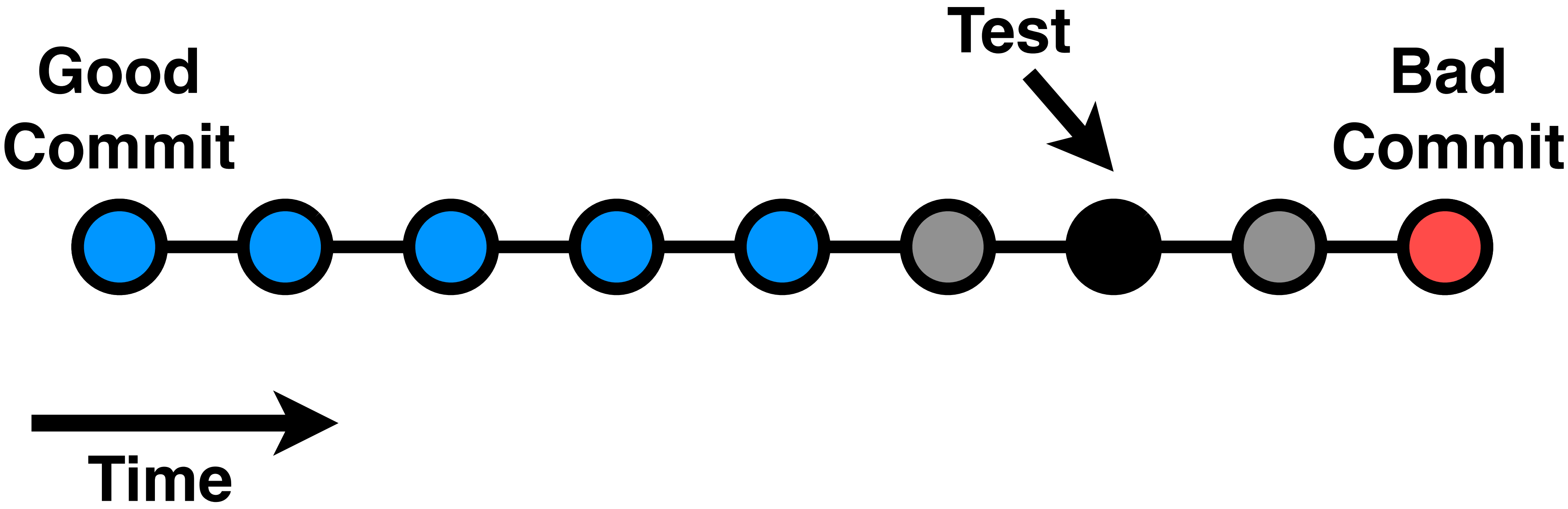

Test
Good Commit
Bad Commit
Time

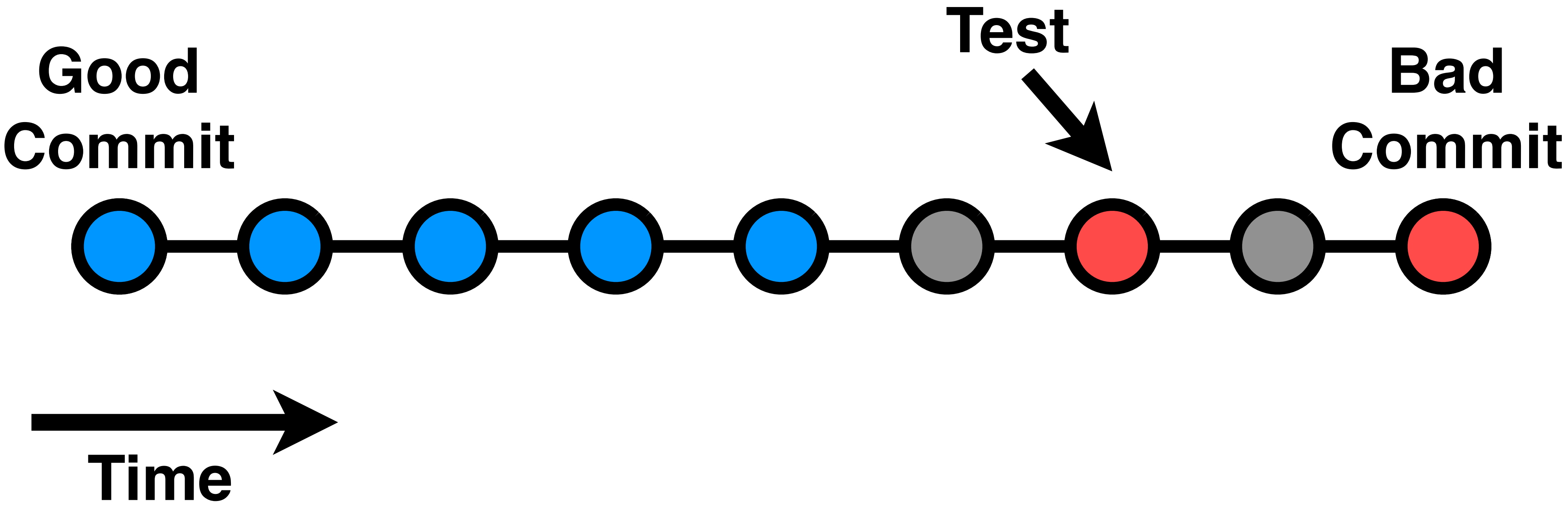


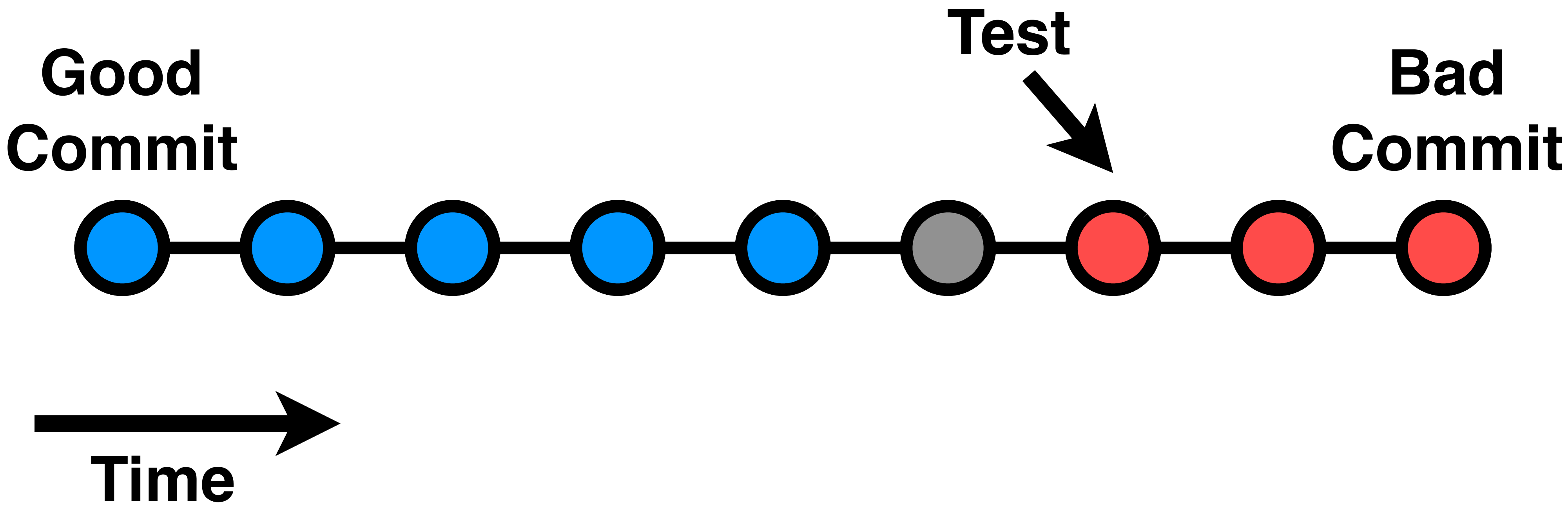
Good Commit

Bad Commit

Time

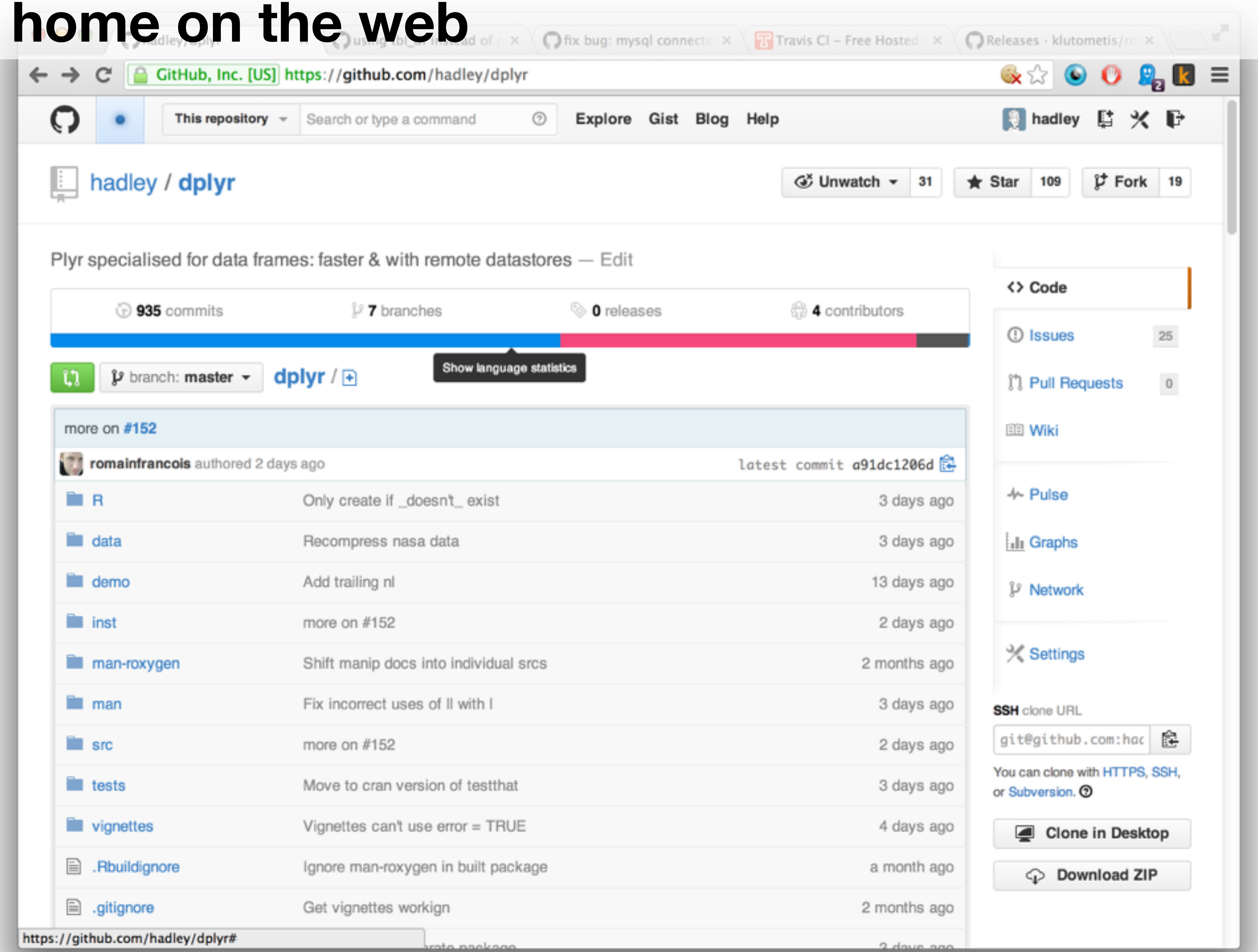

Test
Good Commit
Bad Commit
Time


Test
Good Commit
Bad Commit
Time


Test
Good Commit
Bad Commit
Time

Community

A home on the web



hadley / dplyr

Plyr specialised for data frames: faster & with remote datastores — Edit

935 commits 7 branches 0 releases 4 contributors

branch: master dplyr /

more on #152

romainfrancois authored 2 days ago latest commit a91dc1206d

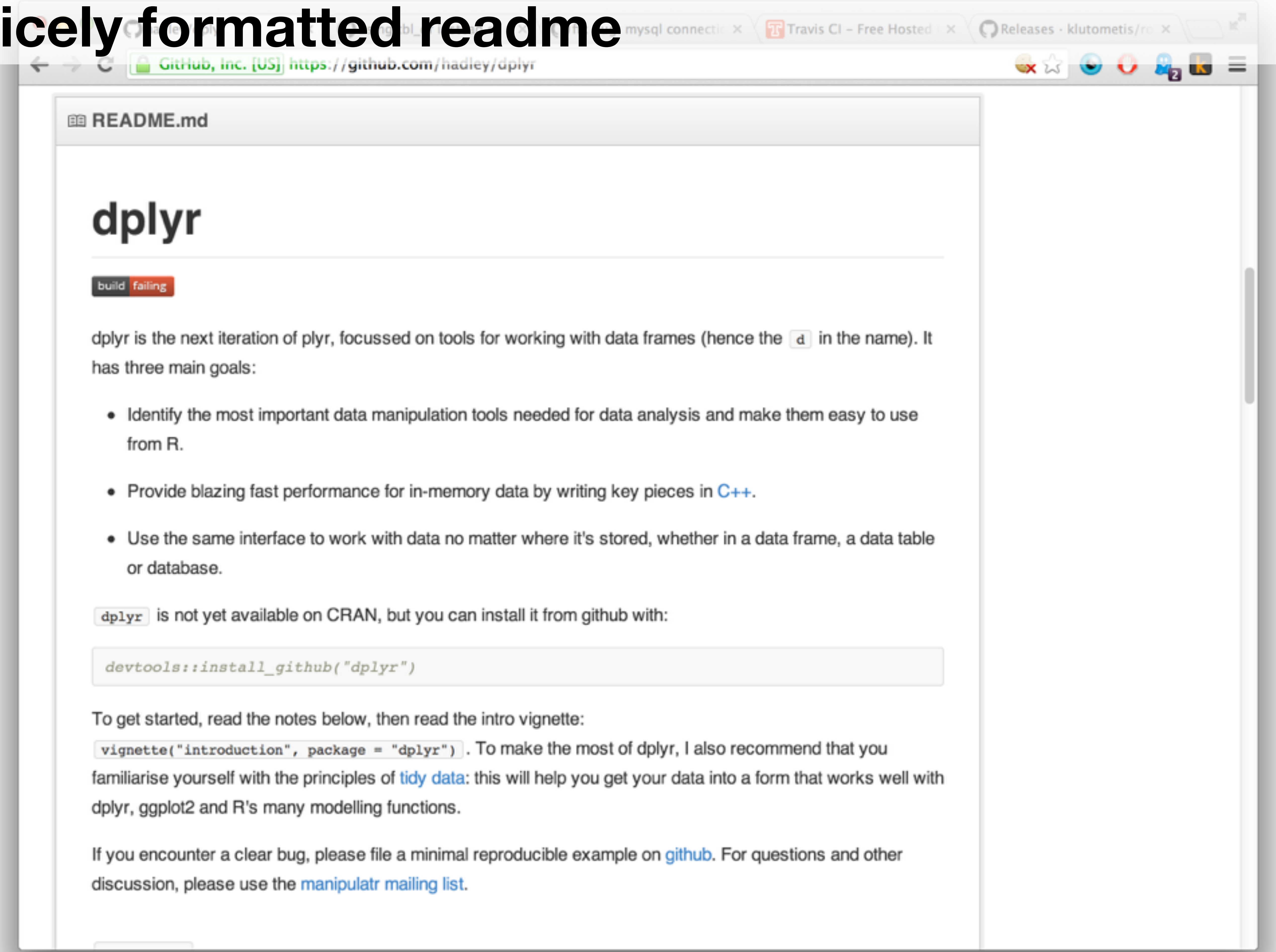
R	Only create if _doesn't_ exist	3 days ago
data	Recompress nasa data	3 days ago
demo	Add trailing nl	13 days ago
inst	more on #152	2 days ago
man-roxygen	Shift manip docs into individual srcs	2 months ago
man	Fix incorrect uses of ll with l	3 days ago
src	more on #152	2 days ago
tests	Move to cran version of testthat	3 days ago
vignettes	Vignettes can't use error = TRUE	4 days ago
.Rbuildignore	Ignore man-roxygen in built package	a month ago
.gitignore	Get vignettes workign	2 months ago

SSH clone URL

git@github.com:hac

You can clone with HTTPS, SSH, or Subversion.

Nicely formatted readme



dplyr

dplyr is the next iteration of plyr, focussed on tools for working with data frames (hence the `d` in the name). It has three main goals:

- Identify the most important data manipulation tools needed for data analysis and make them easy to use from R.
- Provide blazing fast performance for in-memory data by writing key pieces in C++.
- Use the same interface to work with data no matter where it's stored, whether in a data frame, a data table or database.

`dplyr` is not yet available on CRAN, but you can install it from github with:

```
devtools::install_github("dplyr")
```

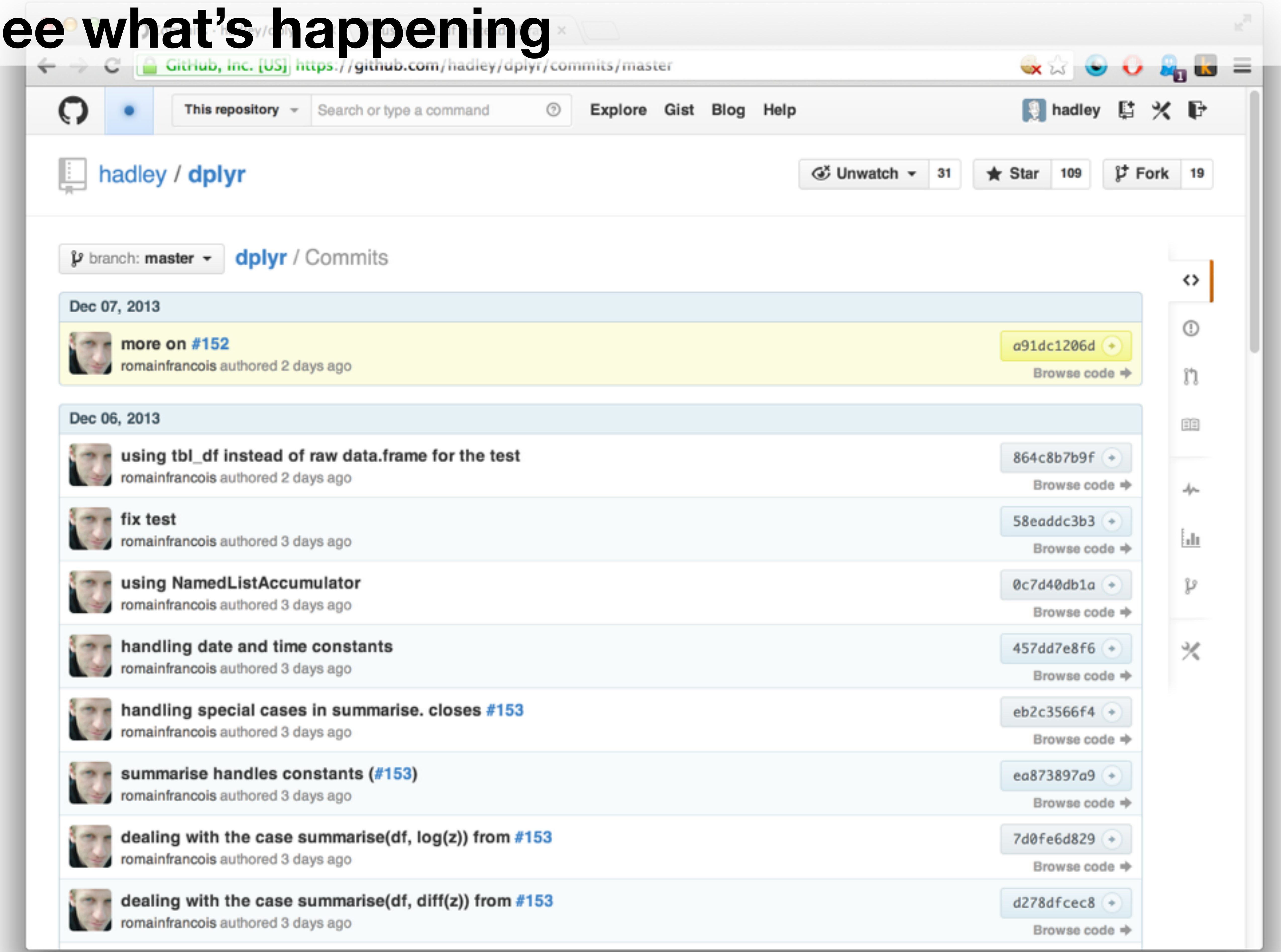
To get started, read the notes below, then read the intro vignette: `vignette("introduction", package = "dplyr")`. To make the most of dplyr, I also recommend that you familiarise yourself with the principles of tidy data: this will help you get your data into a form that works well with dplyr, ggplot2 and R's many modelling functions.

If you encounter a clear bug, please file a minimal reproducible example on github. For questions and other discussion, please use the manipulatr mailing list.

```
# Easy to install a package from github
library(devtools)
install_github(”hadley/devtools”)

# Github is a viable location for package
# distribution, but it doesn’t yet have the
# reach or authority of CRAN. Currently
# about ~3,000 packages on github.
```

See what’s happening



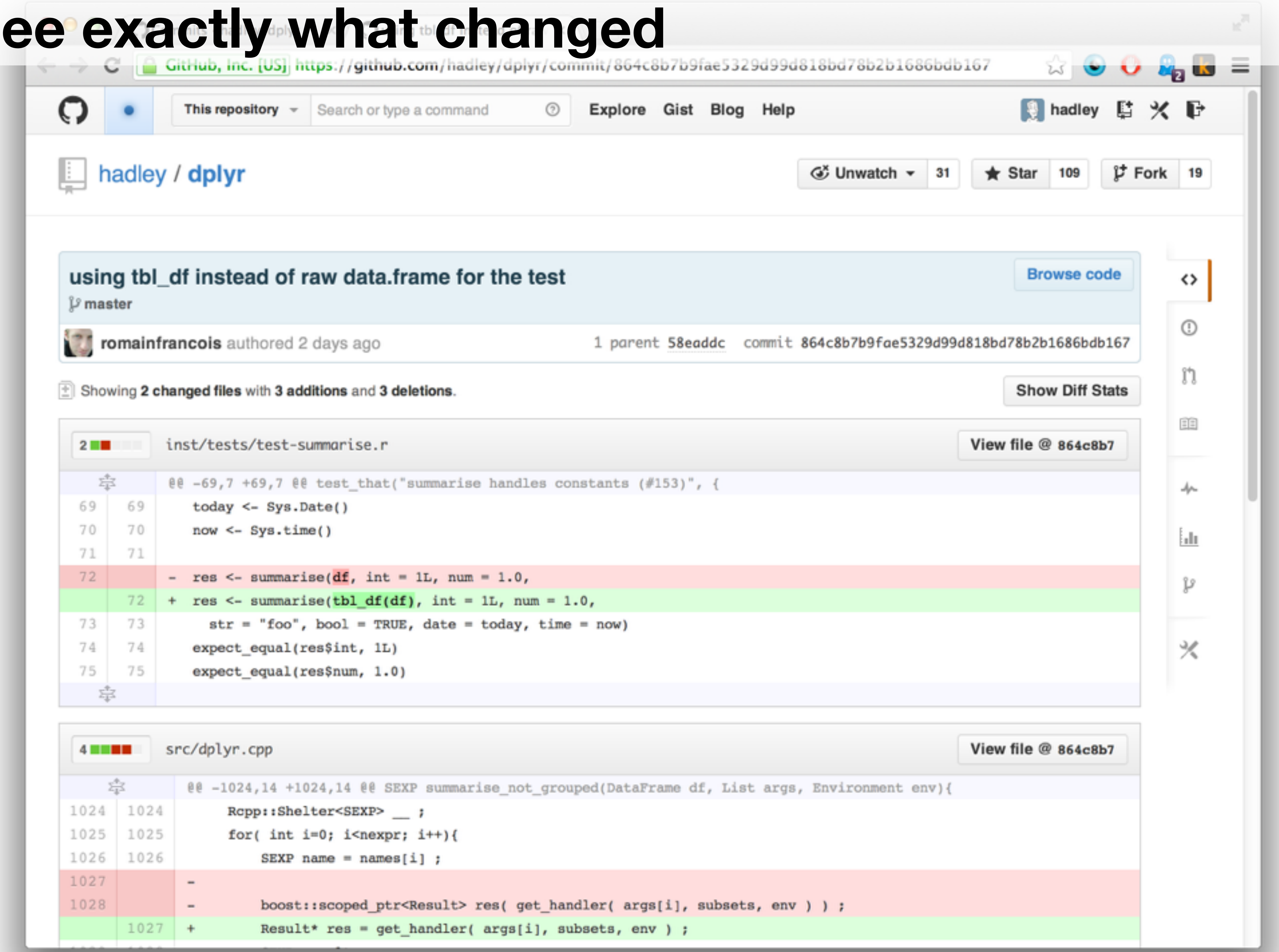
Dec 07, 2013

- more on #152 — romainfrancois authored 2 days ago — a91dc1206d — Browse code

Dec 06, 2013

- using tbl_df instead of raw data.frame for the test — romainfrancois authored 2 days ago — 864c8b7b9f — Browse code
- fix test — romainfrancois authored 3 days ago — 58eaddc3b3 — Browse code
- using NamedListAccumulator — romainfrancois authored 3 days ago — 0c7d40db1a — Browse code
- handling date and time constants — romainfrancois authored 3 days ago — 457dd7e8f6 — Browse code
- handling special cases in summarise. closes #153 — romainfrancois authored 3 days ago — eb2c3566f4 — Browse code
- summarise handles constants (#153) — romainfrancois authored 3 days ago — ea873897a9 — Browse code
- dealing with the case summarise(df, log(z)) from #153 — romainfrancois authored 3 days ago — 7d0fe6d829 — Browse code
- dealing with the case summarise(df, diff(z)) from #153 — romainfrancois authored 3 days ago — d278dfcec8 — Browse code

See exactly what changed



hadley / dplyr

using tbl_df instead of raw data.frame for the test

master

romainfrancois authored 2 days ago 1 parent 58eaddc commit 864c8b7b9fae5329d99d818bd78b2b1686bdb167

Showing **2 changed files** with **3 additions** and **3 deletions**.

2 inst/tests/test-summarise.r

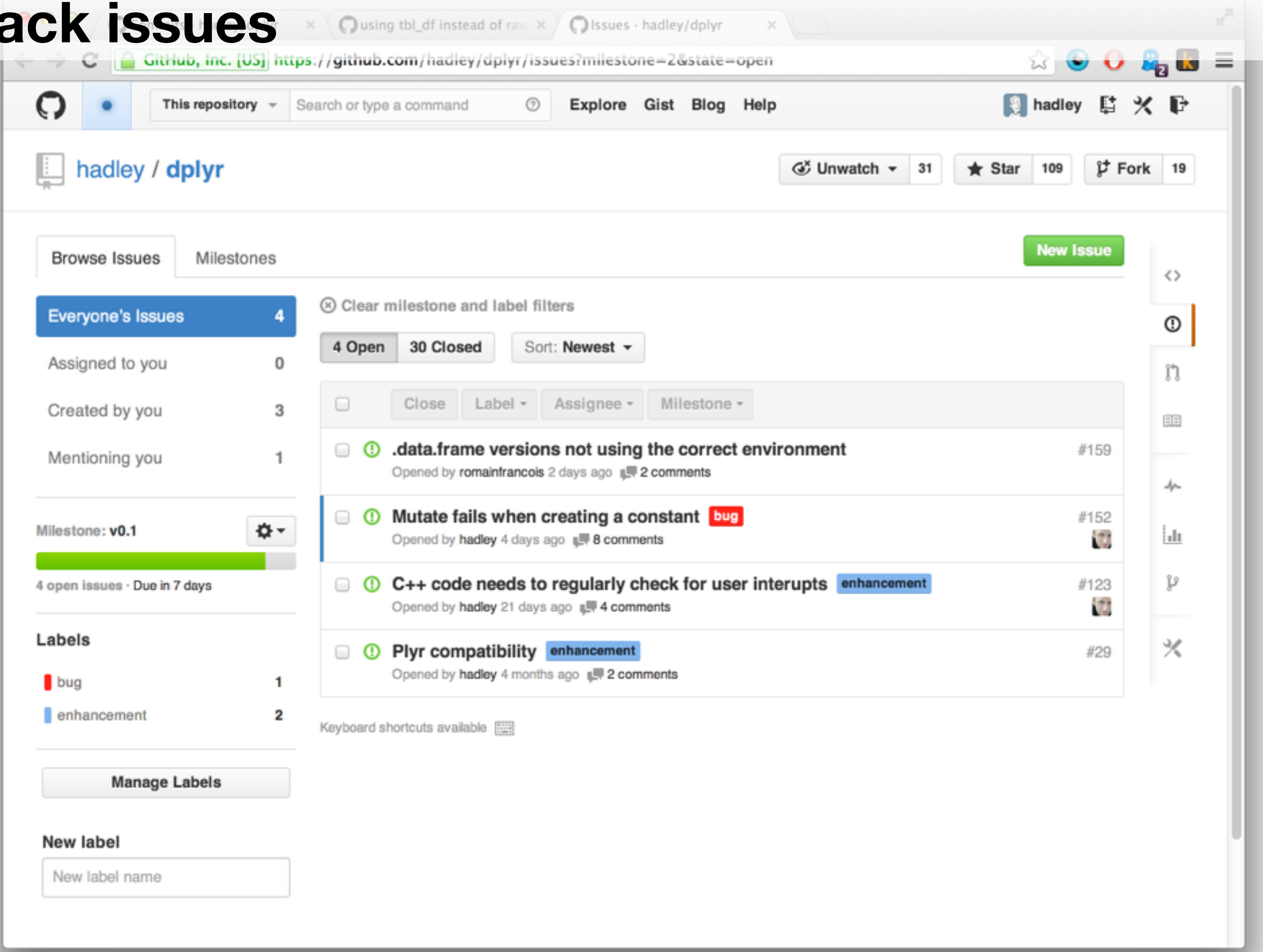
```
@@ -69,7 +69,7 @@ test_that("summarise handles constants (#153)", {
   today <- Sys.Date()
   now <- Sys.time()

-  res <- summarise(df, int = 1L, num = 1.0,
+  res <- summarise(tbl_df(df), int = 1L, num = 1.0,
     str = "foo", bool = TRUE, date = today, time = now)
   expect_equal(res$int, 1L)
   expect_equal(res$num, 1.0)
```

4 src/dplyr.cpp

```
@@ -1024,14 +1024,14 @@ SEXP summarise_not_grouped(DataFrame df, List args, Environment env){
     Rcpp::Shelter<SEXP> __ ;
     for( int i=0; i<nexpr; i++){
         SEXP name = names[i] ;
-
-        boost::scoped_ptr<Result> res( get_handler( args[i], subsets, env ) ) ;
+        Result* res = get_handler( args[i], subsets, env ) ;
```

Track issues



using tbl_df instead of raw ×   Issues · hadley/dplyr ×

GitHub, Inc. [US] https://github.com/hadley/dplyr/issues?milestone=2&state=open

This repository   Search or type a command   Explore   Gist   Blog   Help   hadley

hadley / dplyr   Unwatch 31   Star 109   Fork 19

Browse Issues   Milestones   New Issue

Everyone's Issues	4
Assigned to you	0
Created by you	3
Mentioning you	1

Milestone: v0.1

4 open issues · Due in 7 days

Labels

🟥 bug	1
🟦 enhancement	2

Manage Labels

New label

New label name

Clear milestone and label filters

4 Open   30 Closed   Sort: Newest

□   Close   Label   Assignee   Milestone

□ **.data.frame versions not using the correct environment** #159
Opened by romainfrancois 2 days ago 2 comments

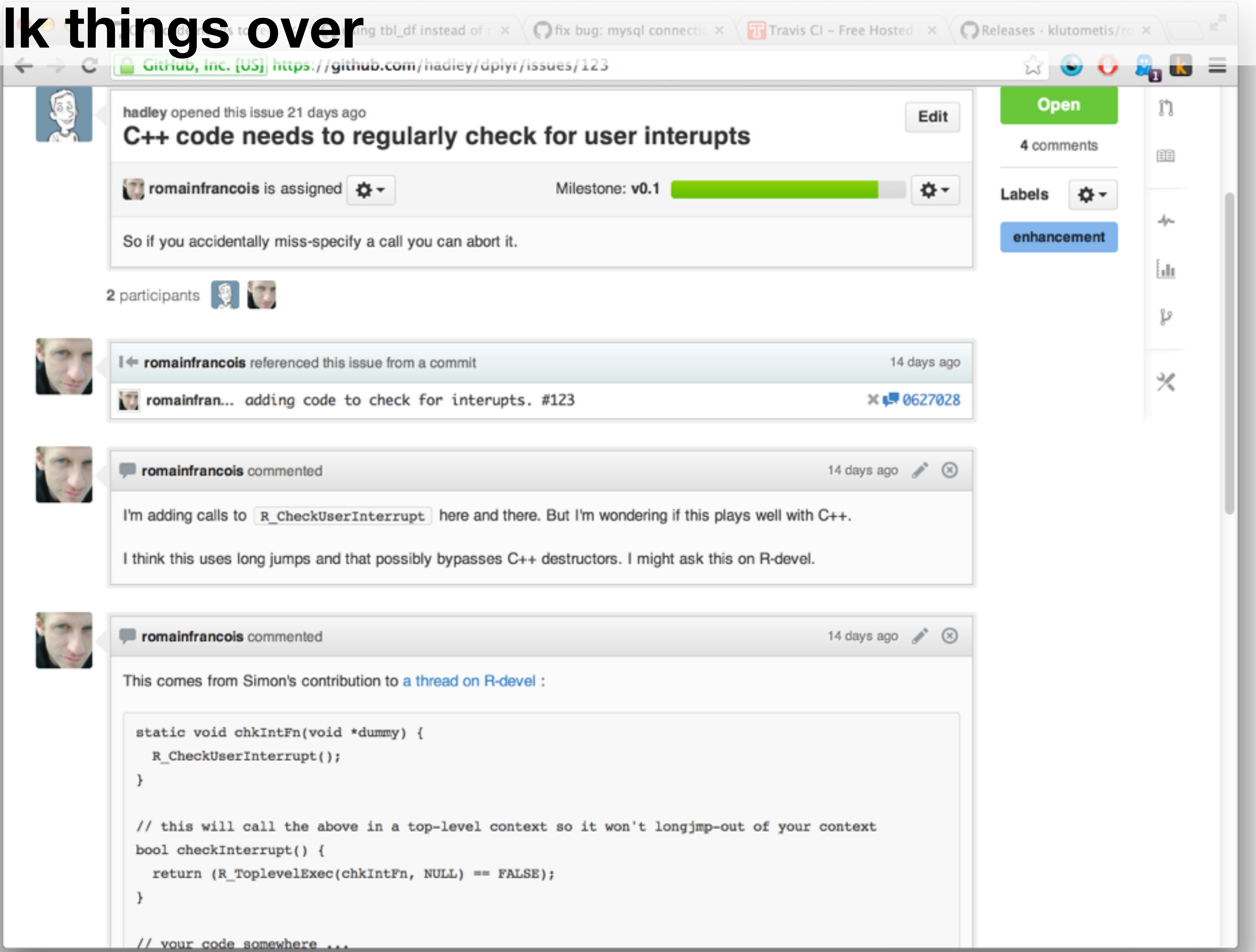
□ **Mutate fails when creating a constant** bug #152
Opened by hadley 4 days ago 8 comments

□ **C++ code needs to regularly check for user interupts** enhancement #123
Opened by hadley 21 days ago 4 comments

□ **Plyr compatibility** enhancement #29
Opened by hadley 4 months ago 2 comments

Keyboard shortcuts available

Talk things over



hadley opened this issue 21 days ago

C++ code needs to regularly check for user interupts

romainfrancois is assigned

Milestone: **v0.1**

So if you accidentally miss-specify a call you can abort it.

2 participants

romainfrancois referenced this issue from a commit 14 days ago

romainfran... *adding code to check for interupts. #123* 0627028

romainfrancois commented 14 days ago

I'm adding calls to `R_CheckUserInterrupt` here and there. But I'm wondering if this plays well with C++.

I think this uses long jumps and that possibly bypasses C++ destructors. I might ask this on R-devel.

romainfrancois commented 14 days ago

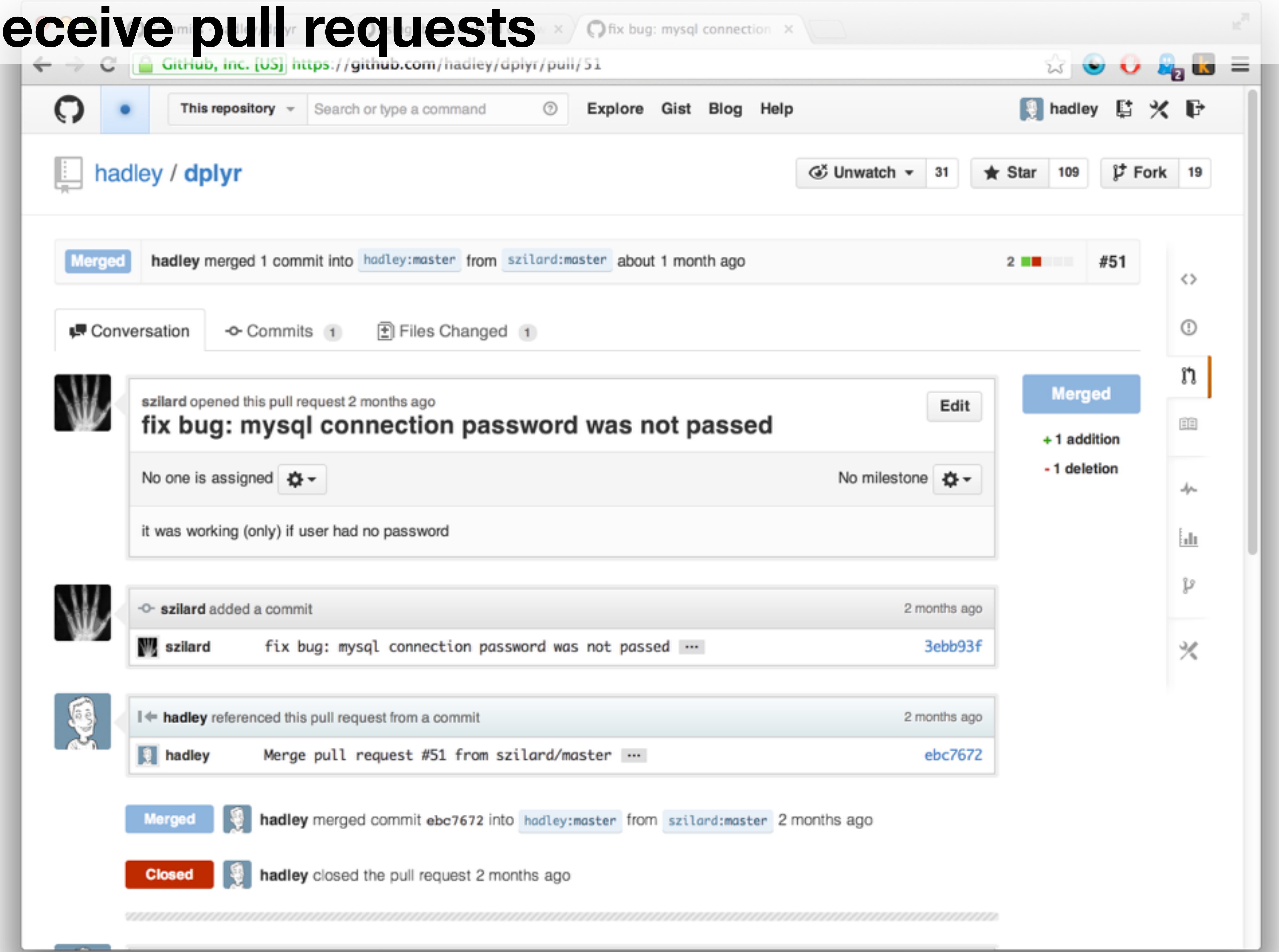
This comes from Simon's contribution to a thread on R-devel :

```
static void chkIntFn(void *dummy) {
  R_CheckUserInterrupt();
}

// this will call the above in a top-level context so it won't longjmp-out of your context
bool checkInterrupt() {
  return (R_ToplevelExec(chkIntFn, NULL) == FALSE);
}

// your code somewhere ...
```

Receive pull requests



hadley / dplyr

Merged hadley merged 1 commit into `hadley:master` from `szilard:master` about 1 month ago 2 #51

Conversation · Commits 1 · Files Changed 1

szilard opened this pull request 2 months ago

fix bug: mysql connection password was not passed

No one is assigned · No milestone

it was working (only) if user had no password

Merged

+ 1 addition

- 1 deletion

szilard added a commit · 2 months ago

szilard `fix bug: mysql connection password was not passed` ··· 3ebb93f

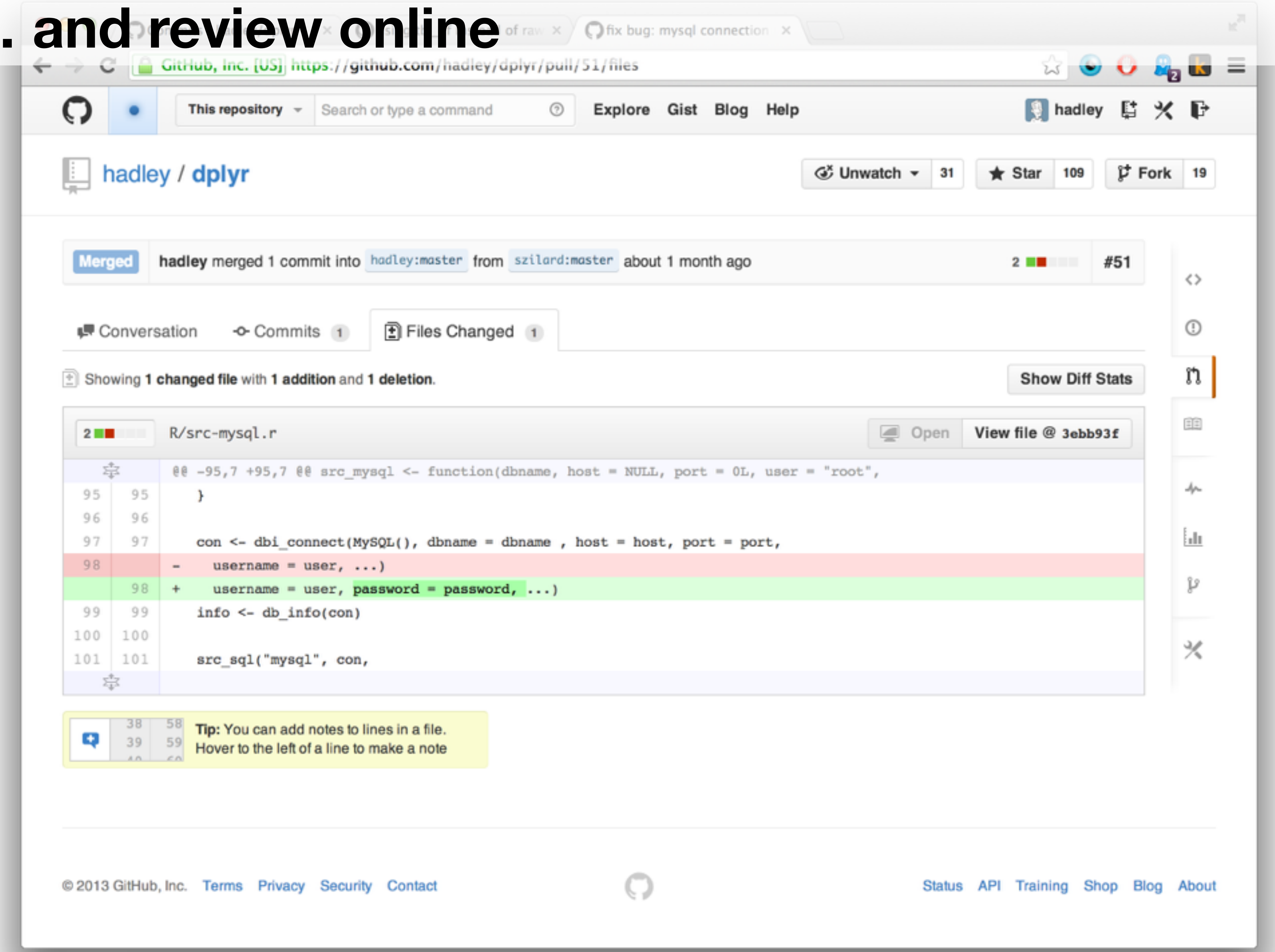
hadley referenced this pull request from a commit · 2 months ago

hadley `Merge pull request #51 from szilard/master` ··· ebc7672

Merged hadley merged commit ebc7672 into `hadley:master` from `szilard:master` 2 months ago

Closed hadley closed the pull request 2 months ago

… and review online



hadley merged 1 commit into `hadley:master` from `szilard:master` about 1 month ago

Showing **1 changed file** with **1 addition** and **1 deletion**.

R/src-mysql.r

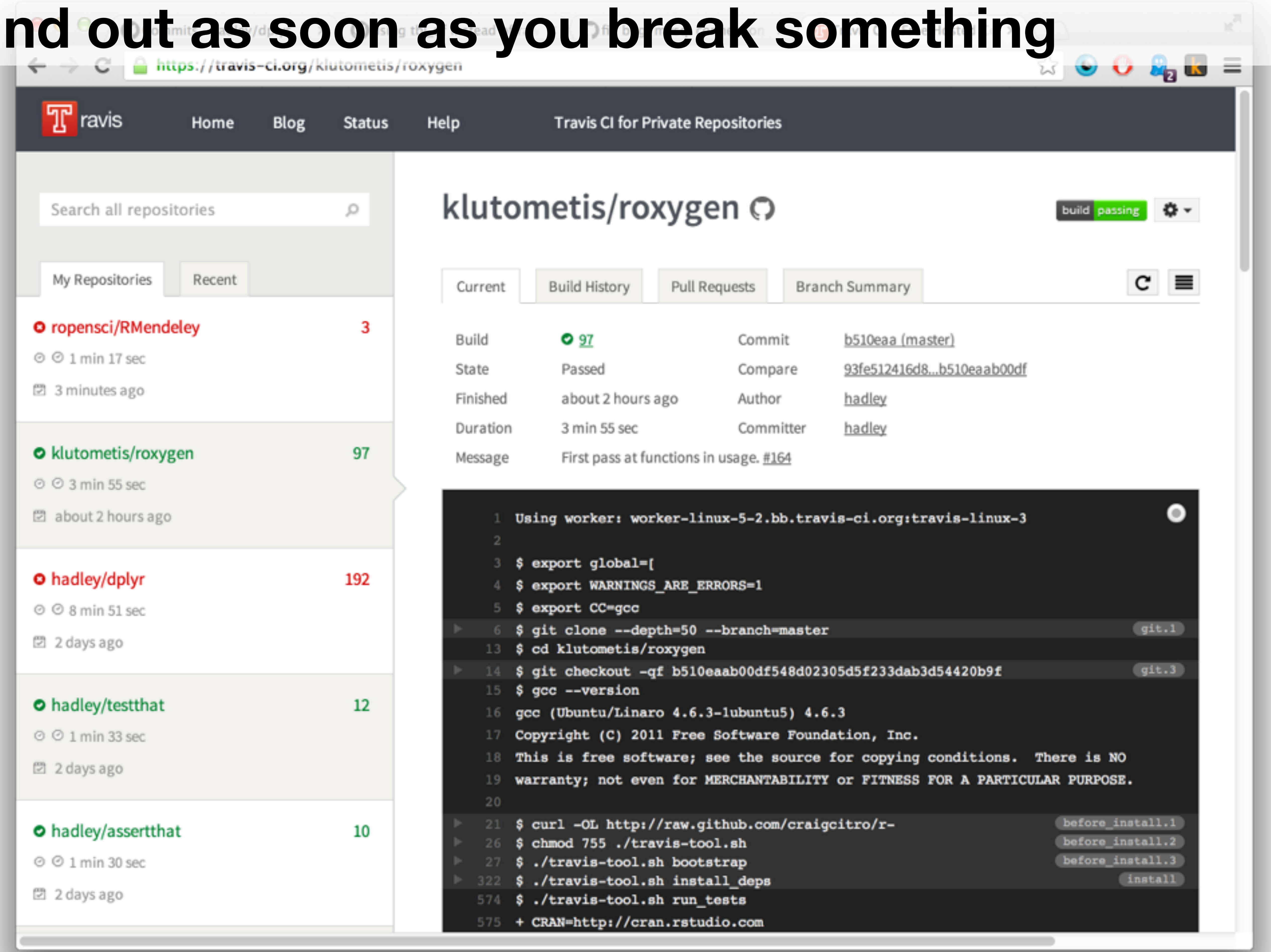
```
@@ -95,7 +95,7 @@ src_mysql <- function(dbname, host = NULL, port = 0L, user = "root",
   }

   con <- dbi_connect(MySQL(), dbname = dbname , host = host, port = port,
-    username = user, ...)
+    username = user, password = password, ...)
   info <- db_info(con)

   src_sql("mysql", con,
```

Tip: You can add notes to lines in a file.
Hover to the left of a line to make a note

Find out as soon as you break something



ropensci/RMendeley 3
1 min 17 sec
3 minutes ago

klutometis/roxygen 97
3 min 55 sec
about 2 hours ago

hadley/dplyr 192
8 min 51 sec
2 days ago

hadley/testthat 12
1 min 33 sec
2 days ago

hadley/assertthat 10
1 min 30 sec
2 days ago

klutometis/roxygen

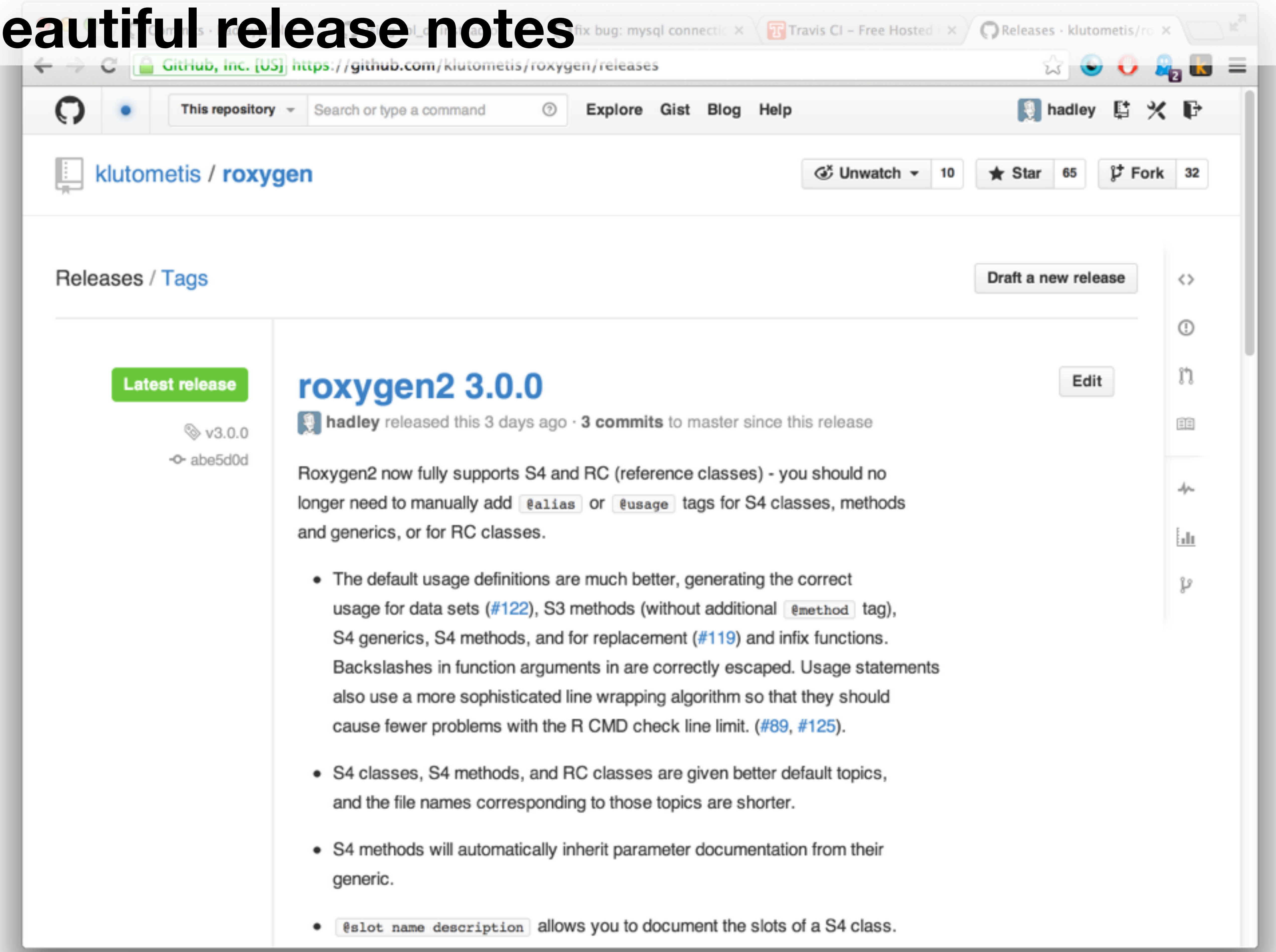
Build	97	Commit	b510eaa (master)
State	Passed	Compare	93fe512416d8...b510eaab00df
Finished	about 2 hours ago	Author	hadley
Duration	3 min 55 sec	Committer	hadley
Message	First pass at functions in usage. #164		

```
Using worker: worker-linux-5-2.bb.travis-ci.org:travis-linux-3

$ export global=[
$ export WARNINGS_ARE_ERRORS=1
$ export CC=gcc
$ git clone --depth=50 --branch=master                                  git.1
$ cd klutometis/roxygen
$ git checkout -qf b510eaab00df548d02305d5f233dab3d54420b9f              git.3
$ gcc --version
gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
Copyright (C) 2011 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

$ curl -OL http://raw.github.com/craigcitro/r-                before_install.1
$ chmod 755 ./travis-tool.sh                                  before_install.2
$ ./travis-tool.sh bootstrap                                  before_install.3
$ ./travis-tool.sh install_deps                                        install
$ ./travis-tool.sh run_tests
+ CRAN=http://cran.rstudio.com
```

Beautiful release notes



roxygen2 3.0.0

hadley released this 3 days ago · **3 commits** to master since this release

Roxygen2 now fully supports S4 and RC (reference classes) - you should no longer need to manually add `@alias` or `@usage` tags for S4 classes, methods and generics, or for RC classes.

- The default usage definitions are much better, generating the correct usage for data sets (#122), S3 methods (without additional `@method` tag), S4 generics, S4 methods, and for replacement (#119) and infix functions. Backslashes in function arguments in are correctly escaped. Usage statements also use a more sophisticated line wrapping algorithm so that they should cause fewer problems with the R CMD check line limit. (#89, #125).
- S4 classes, S4 methods, and RC classes are given better default topics, and the file names corresponding to those topics are shorter.
- S4 methods will automatically inherit parameter documentation from their generic.
- `@slot name description` allows you to document the slots of a S4 class.

Demo

Easily contribute small fixes

https://github.com/mjockers/syuzhet

My advice

Learning Git & Github

- Is frustrating at first & there’s a lot of new vocab to learn.
- Pays off by increasing the pace of your coding because you don't have to worry as much about mistakes.
- Pays off by connecting you with a community of like minded people who can help make your code better.

http://r-pkgs.had.co.nz/git.html